

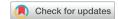
The American Journal of

PATHOLOGY

ajp.amjpathol.org

REVIEW

Pathology Image Analysis Using Segmentation Deep Learning Algorithms



Shidan Wang, * Donghan M. Yang, * Ruichen Rong, * Xiaowei Zhan, * and Guanghua Xiao * † †

From the Quantitative Biomedical Research Center,* Department of Population and Data Sciences, the Department of Bioinformatics,† and the Harold C. Simmons Comprehensive Cancer Center,† University of Texas Southwestern Medical Center, Dallas, Texas

Accepted for publication May 9, 2019.

Address correspondence to Xiaowei Zhan, Ph.D., or Guanghua Xiao, Ph.D., Quantitative Biomedical Research Center, Department of Population and Data Sciences, Harold C. Simmons Comprehensive Cancer Center, UT Southwestern Medical Center, 6000 Harry Hines Blvd., Dallas, TX 75390. E-mail: xiaowei.zhan@utsouthwestern.edu or guanghua.xiao@utsouthwestern.edu.

With the rapid development of image scanning techniques and visualization software, whole slide imaging (WSI) is becoming a routine diagnostic method. Accelerating clinical diagnosis from pathology images and automating image analysis efficiently and accurately remain significant challenges. Recently, deep learning algorithms have shown great promise in pathology image analysis, such as in tumor region identification, metastasis detection, and patient prognosis. Many machine learning algorithms, including convolutional neural networks, have been proposed to automatically segment pathology images. Among these algorithms, segmentation deep learning algorithms such as fully convolutional networks stand out for their accuracy, computational efficiency, and generalizability. Thus, deep learning-based pathology image segmentation has become an important tool in WSI analysis. In this review, the pathology image segmentation process using deep learning algorithms is described in detail. The goals are to provide quick guidance for implementing deep learning into pathology image analysis and to provide some potential ways of further improving segmentation performance. Although there have been previous reviews on using machine learning methods in digital pathology image analysis, this is the first in-depth review of the applications of deep learning algorithms for segmentation in WSI analysis. (Am J Pathol 2019, 189: 1686-1698; https://doi.org/ 10.1016/j.ajpath.2019.05.007)

Optical microscopy of pathology slides captures the histologic details of tissues in high resolution. With the rapid advance of technology, whole slide imaging (WSI) is becoming part of the routine procedure for clinical diagnosis of many diseases. The emergence of digital pathology ^{1,2} provides new opportunities to develop algorithms and software tools that can assist pathologists in clinical diagnosis and researchers in studying disease mechanisms. The digitalized pathology slides are often called images in the computer vision field, and can benefit from many image analysis algorithms. As an example, the common task where pathologists locate and recognize tissue components can also be achieved by image segmentation and recognition algorithms.

Nowadays, digital pathology is making rapid progress owing to the success of deep learning.³ Before the application of deep learning algorithms, digital pathology, due to its high complexity, achieved limited success with laborious

modeling. Since 2012, deep learning has made significant improvements in all image recognition benchmarks. The applications of deep learning algorithms in digital pathology have had remarkable success in traditional pathology tasks. For example, deep learning algorithms achieved performance comparable to pathologists in interpreting whole slide images for the detection of tumor regions and lymph node metastases. Although this comparable performance may not generalize to all task domains, advanced methodology is anticipated to solve or aid in common challenges faced by pathologists, including locating neoplasia within a tissue and quantifying specific features such as mitoses and inflammation.

Supported by NIH grants 1R01GM115473, 5R01CA152301, 5P30CA142543 and 1R01CA172211; and Cancer Prevention and Research Institute of Texas grant RP120732.

Disclosures: None declared.

To understand how deep learning excels in these areas, we build conceptual connections of deep learning in the machine learning literature. In essence, deep learning is a special kind of artificial neural network (ANN), which is one category of machine learning algorithm. Deep learning and other ANNs are inspired by biological neural networks and mathematically construct a network model with multiple connected layers. The first network layer (called the input layer) receives inputs (eg, slide images). It has a set of parameters and can use them to compute outputs. Similarly,

each successive network layer receives inputs from its previous layers, uses its parameters, and computes outputs. At the end, the last network layer (called the output layer) calculates the outputs of the whole model. The layers between the input and output layers are not visible because they do not directly receive model input or generate model outputs, and thus are called the hidden layers. The structure of a segmentation neural network is illustrated in Figure 1A. In this process, prediction outputs from a good neural network can well approximate the observed outputs.

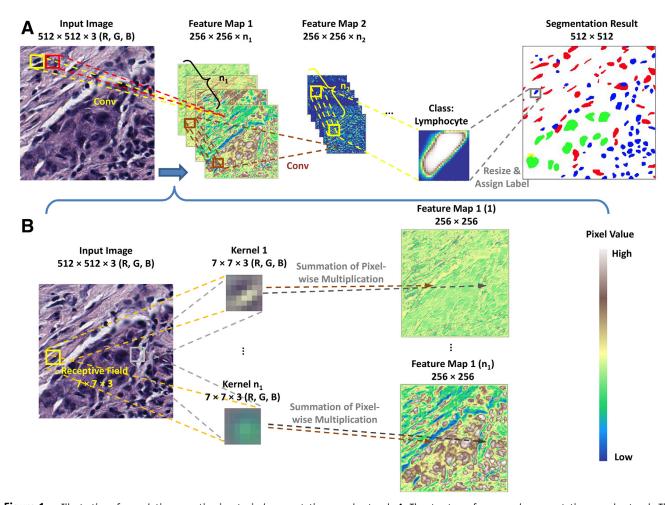


Illustration of convolution operation in a typical segmentation neural network. A: The structure of an example segmentation neural network. The input is a 512 imes 512 imes 3 hematoxylin and eosin—stained lung cancer pathology image from the National Lung Screening Trial (NLST). After the first convolution operation with n1 kernels, a 256 imes 256 imes n1 feature map was generated (the height and width differences between input and feature map 1 are due to stride in the convolution operation, which means the step size of the sliding window is equal to the stride value. In this case, stride = 2. Feature map 1 was then used as input for the second convolution operation with n2 kernels, which yields a $128 \times 128 \times n2$ feature map. In this example, the bounding box, category, and mask of each object is then determined based on feature map 2. Mapping all of the identified objects back to their positions in the input image generates the final segmentation output. In this case, the objects to be identified are cell nuclei, and categories are different cell types. Green: tumor nuclei; blue: lymphocyte nuclei; red: stroma nuclei. B: The first step of the convolution operation shown in A. For illustration purposes, only two of the n1 kernels and corresponding outputs in the first convolutional (Conv) layer are plotted. Each kernel is used to generate one layer ($256 \times 256 \times 1$) of feature map 1. The values of the kernels (middle) are retrieved from a convolutional neural network model trained with pathology images. There are $7 \times 7 \times 3$ (width \times height \times R/G/B channel) = 147 values in each kernel. A receptive field (left) is the part of the image covered by the kernel, and it has the same size (147 values) as the kernel. An output pixel is calculated by summing the products of the pixel-wise multiplications between the values in one receptive field and the corresponding values in the kernel (ie, summation of 147 products). In this illustration, the pixel indicated by the brown arrow in the top panel is calculated from the receptive field surrounded by the yellow rectangle and kernel 1, whereas the pixel indicated by the brown arrow in the bottom panel is calculated from the same receptive field and kernel 2. By sliding the receptive fields along the width and height dimensions, the 2-dimensional output (also called the feature map) can be obtained. In this example, the output of kernel 1 reveals fibrous-like structures whereas the output of kernel n1 reveals nucleilike structures.

Although ANNs claim excellent performances based on theoretical work, ¹¹ historically, it has been notoriously hard to calculate the network parameters when the total number of network layers exceeded three, which limited the performance of the model. Fortunately, this is no longer a severe bottleneck, owing to the advancements in computational hardware, the scale of data accumulation, and the improvements in algorithms. Nowadays, popular ANNs can have hundreds of layers. The machine learning community refers to these algorithms as *deep learning* to distinguish them from the conventional *shallow* ANN algorithm.

In this review, the application of deep learning algorithms in pathology image analysis is the focus. Convolutional neural networks (CNNs) are introduced, which have been widely used for image classification and pathology image analysis, such as tumor region and metastasis detection. In essence, a CNN can have a series of convolution layers as the hidden layers and thus make the network *deep*. This network structure enables the extraction of representational

features for prediction. The design of CNN is inspired by the functional mechanism of the visual cortex³: instead of using all outputs from the previous layer, a convolution kernel only focuses on a certain area, the so-called receptive field, to compute a feature at the corresponding spatial position. By spatially sliding the receptive field along the input dimensions (eg, along the width and height directions for two-dimensional images), a feature map is computed as the outputs from the convolution layer. This process is illustrated in Figure 1B. Because the number of parameters is determined by the size of the receptive field, convolution layers have many fewer parameters than the image size. This design thus effectively reduces the number of parameters within a neural network and greatly improves its computational efficiency.

In addition to image classification, CNNs have also been implemented for pathology image segmentation. To perform image segmentation for large data (eg, whole slide pathology images), the image is first divided into many small patches. A CNN is trained to classify these patches,

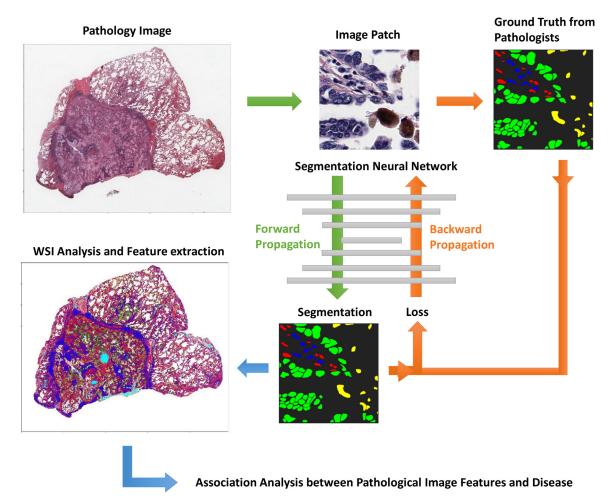


Figure 2 Flow chart of pathology image analysis using segmentation deep learning algorithms. The example pathology image is a hematoxylin and eosin—stained image from the National Lung Screening Trial (NLST). Green arrows are steps that are necessary for both training and application phases; orange arrows are steps that are only performed during the neural network training phase; and blue arrows are steps only performed during the application process. WSI, whole slide image.

and all patches in the same class are combined into one segmented area. Fine spatial resolution of segmentation can be achieved by using patches of small sizes; however, the patches need to be large enough that they can be classified accurately. To overcome the tradeoff between segmentation resolutions and patch size, instead of cutting the image region into adjacent patches, a moving window with a small step size is generally used to generate patches with a certain degree of overlap. In this procedure, the spatial resolution (determined by the step size instead of patch size) is largely improved. However, this demands a substantial amount of computing time and memory, which largely limits the computation speed. In recent years, several deep learning algorithms have been developed specifically for segmentation tasks, which can segment the image at pixel resolution and at a relatively high speed. In this review, the segmentation deep learning algorithms refer to semantic or instance segmentation algorithms, which are derivatives of CNNs. Compared with patch-based CNNs, segmentation deep learning algorithms are more computationally efficient in pixel classification and thus serve as powerful tools to extract detailed image information at pixel resolution. 13,14 Compared with multilabel classification algorithms, which can identify¹⁵ and locate¹⁶ objects of different types, the segmentation deep learning algorithms detect not only the objects but also the segmentation boundaries. Compared with conventional image segmentation algorithms that are not based on deep learning, deep learning algorithms have robust performances under different staining conditions¹⁷ because they do not heavily rely on staining intensity or hand-crafted (ie, manually defined) features, ^{18–22} and they can utilize neighborhood structural information. Thus, deep learning—based segmentations are anticipated to become an important tool in WSI analysis.

In this review, the detailed process of deep learning—based pathology image segmentation is described, including data preparation, image preprocessing, model selection and construction, post-processing, and feature extraction and association with disease (Figure 2). The goals of this review are to provide quick guidance for implementing deep learning—based segmentation for pathology images and to provide some potential ways of further improving the segmentation performance for experienced investigators. Although there have previously been in-depth reviews on using machine learning methods, including deep learning, in digital pathology image analysis, ^{4,23} this is the first review of the implementations of segmentation deep learning algorithms for WSI analysis.

Table 1 Summary of Tools for Mask Preparation

| Software name: description | Support multiclass labeling? | Area selection method | Resource | Active* |
|---|--|---|---|---------|
| Photoshop (Adobe, San Jose, CA): Sophisticated commercial product to draw masks | Yes | Flexible, including polygonal, brush, and filling | https://www.adobe.com/ products/photoshop.html | Yes |
| MATLAB Image Segmenter (MathWorks, Natick, MA): To create binary masks with the help of multiple image processing methods | No | Polygonal | https://www.mathworks.com/ help/images/image-segmentation- using-the-image-segmenter-app.html | Yes |
| QuPath: A Java-based pathology image analysis tool | Yes | Manual labeling; automated cell detection | https://github.com/qupath/qupath | No |
| JS Segment Annotator ²⁴ : A web- based image annotation tool | Yes | Area-based clicking | https://github.com/kyamagu/ js-segment-annotator | No |
| LabelMe ²⁵ : A web-based image annotation tool providing iPhone/iPad (Apple, Cupertino, CA) application | Yes | Polygonal | https://github.com/CSAILVision/ LabelMeAnnotationTool | Yes |
| Labelme: A Python-based image annotation tool | Yes | Polygonal | https://github.com/wkentaro/labelme | No |
| OpenSurfaces ²⁶ : A web-based image annotation tool | Yes, different labels will be assigned for each object | Polygonal | https://github.com/seanbell/ opensurfaces-segmentation-ui | No |
| LabelImg: A Python-based image annotation tool | Yes | Polygonal and brush | https://github.com/lzx1413/ LabelImgTool | Yes |
| Labelbox (San Francisco, CA): A commercial product to annotate images and customize user interface | Yes | Polygonal | https://github.com/labelbox/labelbox | Yes |

^{*}Tools are considered under active development if they were updated after October 2017.

Data Preparation

Pathology images are usually as large as giga-pixels. Because the memory associated with a central processing unit (CPU) or a graphic processing unit (GPU) is often limited, the pathology images should first be chopped into small patches and/or resized, if necessary, to fit the CPU or GPU memory (Figure 2). Common image patch sizes range from 256 \times 256 to 512 \times 512 pixels. The size is often chosen to be divisible by 2 to avoid the trouble of padding for pooling layers. Here, padding means adding pixels to the upper, bottom, left, and right sides of the image, respectively. The value of the added pixels is usually set to 0, which is the so-called zero-padding. Pooling means extracting one representative pixel in each receptive field to reduce the size of the feature map. Max-pooling is the most common pooling method, which uses the maximum value to represent a receptive field. Although, in theory, image segmentation neural networks do not require input images to be of the same size, it is preferable for the images to be cropped into the same size to speed up the algorithm and to fully utilize parallel computations on CPU/GPU. Otherwise, the image patches should be resized or padded into the same size before being fed into the neural network. In this case, zero-padding (adding zero-value pixels to the image boundaries) and symmetric-padding (adding pixels with values symmetrical to the original pixel values along the image boundaries) are commonly used. When the input image patches are padded, simply removing the padding region from the segmentation output can yield a result of the same size as the original image.

Training a neural network for image segmentation is a supervised learning process. Thus, to construct a training set for segmentation, the next step is to manually annotate (ie, label) the ground truth. Pathology expertise is essential in this step. Several tools are available for annotating images, and their features are summarized in Table 1. The annotations are exported into one or a group of image masks, which are single-channel binary images of the same size as the input image patch. The contents of masks can be combined to describe the category of each pixel, which is usually coded as categorical data (eg, a natural number, 0, 1, 2, 3 ...).

Image Preprocessing

To accelerate the training phase and improve model generalizability, image preprocessing is needed right before feeding the image patches into the segmentation neural networks (Figure 2). In preprocessing, image normalization is necessary for both training and application phases, whereas augmentation is necessary only for the training phase.

Image Normalization

Feature normalization is commonly used in the machine learning field to ensure that different features have a

similar effect on the response. For a step-wise gradient descent algorithm,³ which is widely used to train deep-learning models, feature normalization also helps accelerate convergence. There are several common ways to perform image normalization. One option is to simply rescale the pixel value into [0, 1] or [-1, 1]. For example, dividing an 8-bit image by 255 will rescale it to the range [0, 1]. Another option is standardization, which refers to transforming the signal from each image channel into a random variable with mean 0 and variance 1:

$$\tilde{X} = \frac{X - \mu}{\sigma},\tag{1}$$

where μ is the sample mean and σ is the sample variance. Pathology images are usually not stationary. Here, stationary means that a certain channel from each image follows the same pixel value distribution. Therefore, in the standardization step, using the mean and variance of the whole training set instead of using the statistics calculated from each individual image is often recommended.

Augmentation

Because there are millions of parameters to train, using a limited training set will easily cause the neural network to memorize rather than learn how to segment. Image data augmentation is an important step, because it could greatly increase the size of the training set, reduce overfitting, and improve generalizability.

Shape Augmentation

There are multiple image shape augmentation methods. Using the projection matrix is fast and effective. A projective transformation simultaneously enables scaling, translation, rotation, and affine transformations under this formula:

$$\begin{bmatrix} \operatorname{scale} \times \operatorname{aspect\ ratio} \times \cos \theta & -\sin \theta & \operatorname{translation\ in\ } x \\ \sin \theta & \frac{\operatorname{scale} \times \cos \theta}{\operatorname{aspect\ ratio}} & \operatorname{translation\ in\ } y \\ g & h & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$
(2)

In this transformation, θ is the counterclockwise rotation degree; g and h control keystone distortions (an image distortion that distorts the rectangle shape into a trapezoid). After the projective transformation, the position of the pixel (x,y) on the original image will be mapped to (u/w,v/w) on the transformed image. Other transformations include horizontal flipping, vertical flipping, and piecewise affine transformation. ²⁷ If shape

augmentation is used, it is critical to apply the same transformation to both the image and its corresponding mask.

Color Augmentation

Because pathology images may look very different due to different staining conditions and slide thicknesses, it is important to make the deep learning algorithm learn to adapt. One possible solution is to normalize pathology images to a uniform scale. Several pathology image standardization methods have been reported.²⁰ However, they are usually timeconsuming and may diminish some intrinsic information. For example, hematoxylin and eosin—stained renal cell carcinoma pathology images are often classified into eosinophilic and basophilic subtypes, which are prone to be stained by eosin or hematoxylin, respectively, and thus have intrinsically different color distributions.²⁸ A comparatively easier solution in the deep learning context is to use color augmentation to mimic practical differences. By adding a random mean and multiplying a random variation to each channel of each image, the sample size is largely augmented. Thus, the neural network can learn to ignore the systematic biases raised from the pathology slide-making process. It is worth noting that the range, mean, and variance of augmentation parameters should be chosen carefully to reduce distorting image-intrinsic features. Other color augmentation methods include adding Gaussian noise, introducing salt-and-pepper noise, and blurring.

Model Selection and Construction

Software Selection

Although deep learning algorithms can be implemented in C/C++, MATLAB (MathWorks, Natick, MA), R (R Foundation for Statistical Computing, Vienna, Austria), or Julia, Python is still the most commonly used language in the deep learning field. There are several open-source Python libraries to choose from: Caffe,²⁹ TensorFlow,³⁰ Keras (https://keras.io, last accessed March 2019), and PyTorch (https://pytorch.org, last accessed July 2019).³¹ Some

software, such as Aperio GENIE (Leica Biosystems, Buffalo Grove, IL; https://pdfs.semanticscholar.org/28bd/c353 c500d08b67eff9871d7f659c85321696.pdf), also incorporate a machine learning—based segmentation function. These software tools greatly decrease the coding effort, but the cost is that the models are less flexible in regard to both model structure and training phase. Thus, the following parts are based on using a model that can be easily customized to best suit the goal of pathology image analysis.

Model Selection

To best suit the needs of pathology image segmentation, it is important to choose a proper neural network structure. Select a semantic or instance segmentation algorithm and the corresponding loss function (*Loss Function*) should be done first, whereas encoder backbone selection and layer manipulation are necessary only when improving the performance. For a quick start, Python implementations of segmentation deep learning algorithms can be easily found on GitHub, which have been summarized online (eg, *https://github.com/mrgloom/awesome-semantic-segmentation*, last accessed March 2019).

Semantic or Instance Segmentation

Currently, several image segmentation models have been reported. On the basis of the design, these models can be divided into two main categories: semantic segmentation and instance segmentation (Figure 3). It is worth noting that either semantic or instance segmentation can be converted into each other through twisting the models, including changing the prediction target and adding post-processing steps, so the models are classified here according to their original implementation.

The goal of semantic segmentation is to segment image parts with different meanings. The first end-to-end and pixel-to-pixel semantic segmentation neural network is the Fully Convolutional Network (FCN). In FCN, the last fully connected layer in CNN is replaced with a deconvolutional layer to efficiently classify each pixel. The

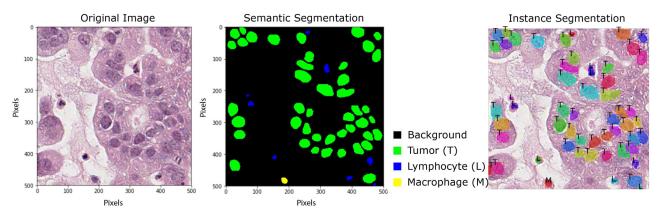


Figure 3 Example of nuclei segmentation in a pathology image. Input is a hematoxylin and eosin—stained image patch (left panel) from the National Lung Screening Trial (NLST). Output is either semantic (middle panel) or instance (right panel) segmentation result. For semantic segmentation, each pixel is assigned a class, whereas for instance segmentation, only objects (nuclei) are picked out and each object is assigned a class.

summation of deconvolutional layers and pooling layers enables FCN to do fine structure segmentation with respect to coarse structure information. Different modifications have been made to FCN to further improve the segmentation performance. For example, U-net greatly increases the number of deconvolutional layers to propagate information to higher resolutions.³² SegNet refines the deconvolutional layers by using indices generated from max-pooling layers.³³ Recently, DeepLab replaced the deconvolutional layers with a fully connected conditional random field (CRF), which is a probabilistic graphical model, and utilized atrous convolution (also called dilated convolution, whose kernel values are interspaced with zeros) and atrous spatial pyramid pooling (a pooling method to combine the outputs from a set of atrous convolution layers with different kernel sizes) instead of standard convolution or pooling layers.³⁴ Other neural network-based semantic segmentation methods have been discussed by Garcia-Garcia et al.³⁵

The goal of instance segmentation is to segment different objects regardless of whether the objects belong to the same category or not. It requires extra effort to recognize object boundaries compared with semantic segmentation. For example, when multiple objects in the same category are close to each other, instance segmentation aims to

distinguish each object with its own boundaries, whereas semantic segmentation has to segment only the combined area of these objects. There are two main strategies of structural design. One is the segment-first strategy, which classifies each object based on the segmentation results.³⁶ The other is the instance-first strategy, which detects the region of interest (ROI) for each instance first. Then, classification and segmentation are applied to the same ROI in parallel. Under this strategy, Mask R-CNN outperforms all other single models on the COCO instance segmentation task as of January 2018.^{37,38}

Encoder Backbone Selection

In addition to the overall model structure design, another important step of model selection is to determine the encoder backbone structure (Figure 4). An encoder generally refers to the part of a neural network that encodes the large inputs to a feature map of much smaller size. According to the Universal Approximation Theorem, a large deep neural network will be able to approximate the arbitrary function the model is trying to learn. However, choosing a proper neural network structure is critical for successful approximation with a practically acceptable convergence rate. A general consensus is that given a

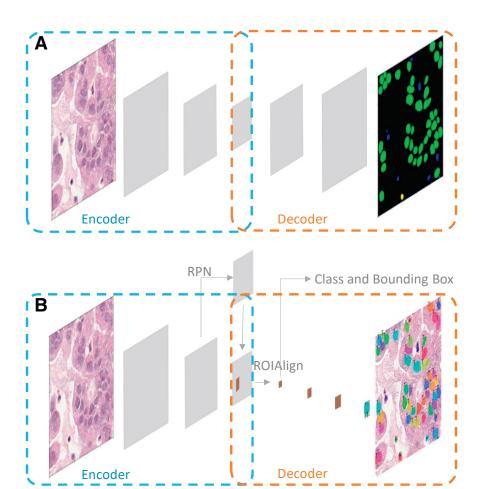


Figure 4 Illustration of encoder and decoder network for semantic segmentation (A) and instance segmentation (B). Specifically, Mask R-CNN structure is used as an example of instance segmentation. ROI, region of interest; ROIAlign, a specific network layer that can efficiently compute feature maps within the candidate regions of interest; RPN, region proposal network, a network to output possible regions (proposal) for objects of interest.

Table 2 Summary of Different Encoder Neural Network Architectures

| Name | Main features | Convolutional Blocks, <i>n</i> * | Convolutional layers in each block, <i>n</i> | Kernel size of convolutional layers, pixels | Dropout layer after each block? | Year |
|---------------------------|--|-------------------------------------|--|---|------------------------------------|------|
| LeNet-5 ⁴² | Established the modern framework of CNN | 2 | 1 | 5 | No | 1998 |
| AlexNet ⁵ | The first to use ReLU as an activation function | 5 | 1 | 3—11 | Yes | 2012 |
| VGG16 ⁴³ | Uses multiple convolution layers with a smaller kernel size to replace a single convolution layer with a large kernel size | 5 | 2—3 | 3 | No | 2014 |
| ResNet101 ⁴¹ | Introduces a deep residual network to solve the gradient vanishing problem and enables deep network architecture | 5 | 1-69 | 1-7 | No | 2015 |
| InceptionV3 ⁴⁴ | Uses Inception modules ⁴⁵ and asymmetric convolutions | 7 | 1—20 | 1—7 | No | 2016 |
| ENet ⁴⁶ | Uses early down-sampling and filter factorization to greatly decrease parameter amount and computation time | 3 | 1—17 | 1—5 | Yes | 2016 |
| DenseNet201 ⁴⁷ | Introduces dense block where each two layers within the block are connected | 6 | 0-97 | 1—7 | No | 2016 |
| ResNetXt101 ⁴⁸ | Splits each block of ResNet into several paths and then aggregates them together, with roughly the same complexity as ResNet. The splitting extent is called cardinality | 5 | 1-69 | 1-7 | No | 2016 |
| Xception ⁴⁹ | Introduces depthwise separable convolution | 6 | 1—26 | 1-3 | No | 2017 |

^{*}A convolutional block is defined as a group of layers from the input layer or the end of the last block before the next sub-sampling layer, which is usually the max-pooling layer or convolutional layer with stride ≥ 2 . When there is branching such as in InceptionV3, the branches are considered as one module until merged. CNN, convolutional neural network; ReLU, rectified linear unit.

similar number of parameters, a deeper network with a smaller kernel size is more efficient than a shallower network with a larger kernel size. 40 Nevertheless, several problems can arise when a network goes deep, such as gradient vanishing and degradation of training accuracy. 41 To efficiently use limited parameters and avoid such problems, multiple combinations of different layer numbers, kernel sizes, and layer manipulation methods have been reported. Some commonly used ones are summarized in Table 2. The architectures are sorted by their date of first proposal to illustrate the evolution history. Although most of these architectures were originally proposed for CNN classification, they can easily be transferred to suit segmentation design by serving as encoders. Although creating a network de novo is always an option, in practice, starting with a published backbone with pre-trained weights is strongly recommended because it largely reduces the training time, requires fewer labeled training samples, and is more likely to perform well. Among the architectures

mentioned, VGG16 and ResNet are the most commonly implemented for both semantic and instance segmentation algorithms, whereas InceptionV3 and ResNet have been reported as useful for pathology image analysis.^{7,50}

Layer Manipulation

After the structure and backbone have been decided, the model can be further regularized and/or refined to improve its stability and generalizability. Regularization methods add a positive number (called a penalty) to the loss function. For example, the penalty term is the summation of the absolute (or squared) values of network parameters in L1 (or L2) regularization, respectively. It has been shown that regularization can reduce the variation in predictions, and thus improve model generalization.⁵¹ A dropout layer can refine a deep learning model by simply setting a random proportion of hidden layer parameters to be zeroes. First proposed in 2012, dropout layers have been widely used to prevent overfitting.^{5,46,52} It is worth noting that dropout is often used

Table 3 Algorithm of Neural Network Training Phase

- 1. **Input**: initial model parameters $W^{(0)}$, loss function L, input I, ground truth GT, gradient descent algorithm A, batch size B, stopping criteria C, training step t=1.
- 2. while C is not meet do
 - 3. Randomly sample B samples, get I_{B} and GT_{B}
 - 4. Forward propagation using $W^{(t-1)}$, yield prediction $P_{B}^{(t)}$
 - 5. Calculate loss $l^{(t)} = L(P_B^{(t)}, GT_B)$
 - 6. Update $W^{(t)} = A(W^{(t-1)}, l^{(t)})$
 - 7. (Optional: update hyperparameters, A and B)
 - 8. t = t + 1
- 9. end while

during the training phase and should be removed in the testing phase. Batch normalization has been proposed and is commonly used to solve the internal covariate shift problem, which is caused by changes in the input distribution during training; thus, it can accelerate the training phase. ⁵³ It can also decrease the dependence on dropout or other regularization layers. Through batch normalization, the input batches are centered and scaled using minibatch mean and variance; the normalized batches are then scaled and shifted using learnable parameters. Other normalization methods include weight normalization. ⁵⁴ and layer normalization, ⁵⁵ but these methods are not as widely used as batch normalization.

Loss Function

The loss function of a deep learning network quantifies the difference between the neural network output and desired behavior (such as the segmentation results or classifications curated by pathologists) given the network parameters. The training phase is essentially a process to minimize loss by adjusting the network parameters iteratively. For example, the partial derivatives of loss to network parameters can be used to update the network parameters themselves. For semantic segmentation, the most common loss function is pixel-wise cross-entropy between the network outputs and the true segmentation annotations. To further improve the segmentation for adjacent objects (eg, touching cells), Ronneberger et al³² proposed to upweight the cross-entropy loss for the pixels on the separation border. For instance segmentation, the loss function varies among different strategies and often summates losses from multiple tasks. ^{37,56} For the instance-first strategy, ^{37,56} the losses are composed of three parts: categorical classification crossentropy, bound-box regression L1 loss,⁵⁷ and pixel-wise binary cross-entropy. They respectively measure whether the model can classify the images within bound-boxes accurately, whether the bound-boxes locate the objects of interest, and whether the segmentation of images is accurate. It is important to note that when the data sets are unbalanced, which means classes within the data set have unequal proportions, adding class weights to the categorical loss function is one common method to avoid the algorithm predicting all pixels as the major class. The class weights

can be determined through inverse class probability weighting, ⁴⁶ although some reports found weighting was less important when there was only mild unbalance. ¹³ Other methods to balance classes include oversampling and synthetic sampling [eg, SMOTE (synthetic minority oversampling technique)] to enlarge the sample size of the minor class. ⁵⁸

Training Phase

After data preparation and model construction, the next step is to train the segmentation neural network. The training phase is a process to update model parameters and is composed of alternating forward and backward propagations (Figure 2). Forward propagation uses network parameters to compute model outputs (eg, predictions), whereas backward propagation updates the network parameters according to their partial derivatives of the loss function. One forward and one backward propagation construct one training step (called an iteration). The loss will converge to a local minimum through a series of training steps. In practice, although popular Python libraries such as Keras have encapsulated the training phase into a single function, it is meaningful to understand the details of the training phase (Table 3).

Because the loss function of neural networks is usually not convex, properly initializing the model parameters (termed as weight-initialization) is important to improve the convergence and accelerate the training phase. Because many weights trained on large data sets such as ImageNet and on the backbones mentioned above are publicly available, in practice, the best choice is choosing a commonly used backbone and using these pre-trained weights as initial parameters, which is known as transfer learning. 13,56 If pretrained weights are not available, the parameters should be randomly initialized. Autoencoder, which aims to reconstruct the inputs, can then be used to train the network (referred to as unsupervised pre-training) before supervised training ^{59,60}; because no ground truth labeling is required in this step, this pre-training strategy helps to improve model performance by utilizing unlabeled data.

Selecting proper hyperparameters also helps to accelerate the training phase and improve prediction accuracy. Here, hyperparameters include the gradient descent algorithm, learning rate, momentum, learning rate decay, batch size, and number of epochs to train. Stochastic Gradient Descent randomly samples a batch of images and updates the weights in each step, ⁶¹ and serves as a base for other algorithms such as ADAM ⁶² and Nesterov Accelerated Gradient. ⁶³ The learning rate is often empirically determined and is tuned according to the learning curve. Usually, it helps the convergence to set a relatively high learning rate at first and decrease it later. Batch size refers to the number of images used in each training step and is usually limited by CPU or GPU memory; higher batch sizes often lead to a more stable learning process. Limiting the number of

training epochs can help avoid the overfitting problem, but this is not critical when the training set is large or with enough data augmentation. One way to automatically stop the training phase is to monitor the loss history of the validation set and stop the training when the loss fails to decrease in the validation set after a sequence of 10 or more epochs. Although empirically selecting hyperparameters often gives satisfactory results, several automated hyperparameter search methods can be implemented to optimize hyperparameters, including grid search, 64 random search, 65 Bayesian optimization, and Bayesian optimization and Hyperband.⁶⁶ Here, grid search looks for the best hyperparameter set from all combinations of specified hyperparameters; random search uses random combinations instead; Bayesian optimization utilizes the performance of all historical hyperparameter sets to determine the next set to try; and Bayesian optimization and Hyperband accelerates the sequential hyperparameter evaluation process of Bayesian optimization through the Hyperband strategy, which assigns limited resource (eg, training epochs) to each training procedure and is easy to parallelize.

Post-Processing of Segmentation Results

Sometimes post-processing of segmentation results can be implemented and further reduce prediction errors. Pixellevel prediction errors can happen either randomly or due to the intrinsic limitations of the neural network models. For example, for ROI-based instance segmentation, the segmentation mask from the model is always within the predicted bounding box. If the predicted bounding box is smaller than an object, the computed segmentation mask is also smaller and results in clipped masks in the border areas. Those errors can be reduced through post-processing. For example, the aforementioned CRF can model the spatial interactions among neighboring pixels⁶⁷ and has been shown to improve image segmentation.³⁴ Based on the knowledge that pixels with similar intensities tend to belong to the same category, CRF designed a specific loss function that originated from the concept of energy in physics. When the loss function is reduced by grouping pixels of similar intensities, the ROI-based segmentation masks can be expanded to recover the clipped areas. 68 Other basic image processing methods, such as filtering out objects according to their area and shape (eg, filtering out a segmentation of cells with extremely small areas), can also help exclude false positives.

Case Studies on Disease-Associated Pathology Image Segmentation

Pathology image segmentation results can facilitate pathologists in locating ROI quickly. By extracting features from segmented regions and correlating these features with pathologic, genomic, and clinical variables, systematic

analysis can provide additional ways to discover clinically meaningful features and to aid in diagnosis and treatment planning. Three case studies are provided to illustrate the process as follows. Some of the following studies use non-deep learning—based segmentation methods or patch-wise CNNs, which cannot achieve the same spatial resolution as FCN or other segmentation algorithms. However, because their analytic methods for correlating pathology images with molecular features are valuable for guiding segmentation analysis in the future, they are also included in this session.

Cancer Vasculature

With developments in computational power and imaging techniques, there is an increasing interest in quantifying the tumor microenvironment and correlating the structural information with tumor invasion and patient prognosis. Vasculature plays a key role in tumor growth and metastasis. In lung cancer adenocarcinoma, an FCN model was trained to segment blood vessels automatically in hematoxylin and eosin—stained slides. This model achieved 0.952 pixel-wise accuracy in the testing set, and could be applicable for pathology images of kidney and breast cancer. Microvessel density was then evaluated and found to be positively correlated with survival.

Nuclei Segmentation in Tumor

Mapping and quantification of nuclei in the tumor area is another significant direction in dissecting the tumor microenvironment. To accurately segment nuclei, especially those that are clustered together in pathology images, Zhang et al⁷⁰ designed a novel segmentation neural network structure, which could simultaneously detect objects (instance segmentation) and classify each image pixel (semantic segmentation). Their strategy improved segmentation of regions of overlapped objects. Graham et al¹ designed another neural network structure, XY-Net, to tackle the nuclei segmentation challenge in pathology image. They added an innovative branch, XY branch, to the basic semantic segmentation structure. For each pixel within the input image, the XY branch could generate the distance to the center of the nuclei where the pixel was located on the two dimensions (x and y) separately. This way, connected nuclei could be distinguished and segmented apart.

Biomarker Quantification

Immunohistochemistry staining is used to localize and quantify biomarkers for a certain cell type or functional state. To automate immunohistochemistry signal quantification, an FCN has been proposed to segment cervical intraepithelial neoplasia pathology images into biomarker-stained and nonstained regions. ¹⁷ In this case, the tissue is costained with p16 (a diagnostic biomarker for cervical intraepithelial neoplasia), Ki-67 (a cancerous biomarker),

and hematoxylin. The segmented ROIs were further divided into multiple layers according to the distance between the basal membrane and the top of the epithelium. Through directly counting the percentage of positive biomarkers in each layer, a proliferation pattern is easily visualized that shows good accordance with the literature.⁷²

Future Directions

One important direction is to improve the segmentation performance. Traditional segmentation networks such as FCN and SegNet do not model the spatial relationships among different objects, which hampers the segmentation performance. As an example, the spatial relationship can be the intrinsic properties of cell components, eg, the cell nuclei must be surrounded by cytoplasm. Several methods have been suggested to incorporate such relationships. CRF explicitly models the spatial relationship as a model prior, and it improves label consistency where the visual appearances are similar. Adaptive affinity fields bring in a loss term to reflect the grouping force for the same category and the separation force for different categories.

Another significant challenge is to accelerate the gigapixel image processing speed. As mentioned in the data preparation part, image patches are extracted from the whole slide and usually analyzed in a sliding-window pattern. Accounting for the time spent on data input/output between hard disks and CPU/GPU as well as the calculation for the forward propagation processes, a whole pathology slide could take as long as 1 hour. One way to speed this up is to use a larger window size, if the GPU (or CPU) memory affords. Another way is to improve the sliding-window pattern by mimicking a pathologist's routine. Pathologists usually first look at the whole slide at a low resolution to locate the candidate ROIs. They then zoom in to examine these ROIs at a higher resolution. This pyramidal examination pattern, if adapted in segmentation algorithms, would save a lot of time compared with the sliding-window pattern.

References

- Jara-Lazaro AR, Thamboo TP, Teh M, Tan PH: Digital pathology: exploring its applications in diagnostic surgical pathology practice. Pathology 2010, 42:512-518
- Webster JD, Dunstan RW: Whole-slide imaging and automated image analysis: considerations and opportunities in the practice of pathology. Vet Pathol 2014, 51:211–223
- 3. LeCun Y, Bengio Y, Hinton G: Deep learning. Nature 2015, 521: 436-444
- Janowczyk A, Madabhushi A: Deep learning for digital pathology image analysis: a comprehensive tutorial with selected use cases. J Pathol Inform 2016, 7:29
- Krizhevsky A, Sutskever I, Hinton GE: ImageNet classification with deep convolutional neural networks. Commun ACM 2017, 60:84

 –90
- Goodfellow I, Bengio Y, Courville A, Bengio Y: Deep Learning. Cambridge, MA, MIT Press, 2016

- Liu Y, Gadepalli K, Norouzi M, Dahl GE, Kohlberger T, Boyko A, Venugopalan S, Timofeev A, Nelson PQ, Corrado GS: Detecting cancer metastases on gigapixel pathology images. arXiv 2017. arXiv: 1703.02442
- Wang D, Khosla A, Gargeya R, Irshad H, Beck AH: Deep learning for identifying metastatic breast cancer. arXiv 2016. arXiv:1606.05718
- Wang S, Chen A, Yang L, Cai L, Xie Y, Fujimoto J, Gazdar A, Xiao G: Comprehensive analysis of lung cancer pathology images to discover tumor shape and boundary features that predict survival outcome. Sci Rep 2018, 8:10393
- Ehteshami Bejnordi B, Veta M, van Diest PJ, van Ginneken B, Karssemeijer N, Litjens G, et al: Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA 2017, 318:2199–2210
- Cybenko G: Approximation by superpositions of a sigmoidal function. Math Control Signal Syst 1989, 2:303

 –314
- Xu J, Luo X, Wang G, Gilmore H, Madabhushi A: A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. Neurocomputing 2016, 191:214–223
- Shelhamer E, Long J, Darrell T: Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 2017, 39:640-651
- Yi F, Yang L, Wang S, Guo L, Huang C, Xie Y, Xiao G: Microvessel prediction in H&E stained pathology images using fully convolutional neural networks. BMC Bioinformatics 2018, 19:64
- 15. Wang J, Yang Y, Mao JH, Huang ZH, Huang C, Xu W: CNN-RNN: a unified framework for multi-label image classification. Edited by Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA: IEEE Computer Society, 2016. pp. 2285–2294
- Chen T, Wang Z, Li G, Lin L: Recurrent attentional reinforcement learning for multi-label image recognition. arXiv 2017. arXiv: 1712.07465
- Sheikhzadeh F, Ward RK, van Niekerk D, Guillaud M: Automatic labeling of molecular biomarkers of immunohistochemistry images using fully convolutional networks. PLoS One 2018, 13: e0100783
- Ruiz A, Kong J, Ujaldon M, Boyer K, Saltz J, Gurcan M: Pathological image segmentation for neuroblastoma using the GPU. Proc IEEE Int Symp Biomed Imaging 2008, 2008:296–299
- Kong H, Gurcan M, Belkacem-Boussaid K: Partitioning histopathological images: an integrated framework for supervised color-texture segmentation and cell splitting. IEEE Trans Med Imaging 2011, 30: 1661–1677
- Alsubaie N, Trahearn N, Raza SEA, Snead D, Rajpoot NM: Stain deconvolution using statistical analysis of multi-resolution stain colour representation. PLoS One 2017, 12:e0169875
- Ma Z, Shiao SL, Yoshida EJ, Swartwood S, Huang F, Doche ME, Chung AP, Knudsen BS, Gertych A: Data integration from pathology slides for quantitative imaging of multiple cell types within the tumor immune cell infiltrate. Diagn Pathol 2017, 12:69
- Gonzalez RC, Woods RE: Digital Image Processing. Upper Saddle River, NJ, Pearson Education, 2002
- Komura D, Ishikawa S: Machine learning methods for histopathological image analysis. Comput Struct Biotechnol J 2018, 16:34

 –42
- Tangseng P, Wu Z, Yamaguchi K: Looking at outfit to parse clothing. arXiv 2017. arXiv:1703.01386
- Russell BC, Torralba A, Murphy KP, Freeman WT: LabelMe: a database and web-based tool for image annotation. Int J Comput Vis 2008, 77:157–173
- Bell S, Upchurch P, Snavely N, Bala K: OpenSurfaces: a richly annotated catalog of surface appearance. ACM Trans Graph 2013, 32.
 Article 11
- Pitiot A, Malandain G, Bardinet E, Thompson PM: Piecewise affine registration of biological images. Biomed Image Registration 2003, 2717:91–101

- Lopez-Beltran A, Scarpelli M, Rodolfo M, Kirkali Z: 2004 WHO classification of the renal tumors of the adults. Eur Urol 2006, 49: 798–805
- Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T: Caffe: convolutional architecture for fast feature embedding. Edited by Proceedings of the 22nd ACM International Conference on Multimedia. New York, NY: ACM, 2014. pp. 675–678
- Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M: Tensorflow: a system for largescale machine learning. Edited by Proceedings of the 12th USE-NIX Symposium on Operating Systems Design and Implementation (OSDI '16). Berkeley, CA: USENIX Association, 2016. pp. 265–283
- Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A: Automatic differentiation in PyTorch. 31st Conference on Neural Information Processing Systems (NIPS 2017), December 4–9, 2017, Long Beach, CA
- Ronneberger O, Fischer P, Brox T: U-Net: Convolutional Networks for Biomedical Image Segmentation. Edited by Navab N, Hornegger J, Wells W, Frangi A. In Medical Image Computing and Computer-Assisted Intervention — MICCAI 2015: Lecture Notes in Computer Science, vol 9351. Cham, Switzerland: Springer, 2015. pp. 234—741
- Badrinarayanan V, Kendall A, Cipolla R: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 2017, 39:2481

 –2495
- Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL: Deep-Lab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 2018, 40:834

 –848
- Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J: A review on deep learning techniques applied to semantic segmentation. arXiv 2017. arXiv:1704.06857
- 36. Liu S, Jia JY, Fidler S, Urtasun R: SGN: sequential grouping networks for instance segmentation. Edited by Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 3516–3524
- He KM, Gkioxari G, Dollar P, Girshick R: Mask R-CNN. Edited by Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 2980–2988
- 38. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick CL: Microsoft COCO: common objects in context. Edited by Fleet D, Pajdla T, Schiele B, Tuytelaars T. In Computer Vision ECCV 2014: Lecture Notes in Computer Science, vol 8693. Cham, Switzerland: Springer, 2014. pp. 740–755
- 39. Hornik K, Stinchcombe M, White H: Multilayer feedforward networks are universal approximators. Neural Networks 1989, 2:359–366
- Yuan Y, Lo YC: Improving dermoscopic image segmentation with enhanced convolutional-deconvolutional networks. IEEE J Biomed Health Inform 2019, 23:519

 –526
- He KM, Zhang XY, Ren SQ, Sun J: Deep residual learning for image recognition. Edited by Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA: IEEE Computer Society, 2016. pp. 770–778
- Lecun Y, Bottou L, Bengio Y, Haffner P: Gradient-based learning applied to document recognition. Proc IEEE 1998, 86:2278–2324
- Simonyan K, Zisserman A: Very deep convolutional networks for large-scale image recognition. arXiv 2014. arXiv:1409.1556
- 44. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z: Rethinking the Inception architecture for computer vision. Edited by Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA: IEEE Computer Society, 2016. pp. 2818–2826
- 45. Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A: Going deeper with

- convolutions. Edited by Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA: IEEE Computer Society, 2015. pp. 1–9
- Paszke A, Chaurasia A, Kim S, Culurciello E: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv 2016. arXiv:1606.02147
- Huang G, Liu Z, van der Maaten L, Weinberger KQ: Densely connected convolutional networks. Edited by Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 2261–2269
- 48. Xie SN, Girshick R, Dollar P, Tu ZW, He KM: Aggregated residual transformations for deep neural networks. Edited by Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 5987–5995
- Chollet F: Xception: deep learning with depthwise separable convolutions. Edited by Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 1800–1807
- 50. Sun JM, Binder A: Comparison of deep learning architectures for H&E histopathology images. Edited by Proceedings of the 2017 IEEE Conference on Big Data and Analytics (ICBDA). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 43–48
- 51. Krogh A, Hertz JA: A simple weight decay can improve generalization. Edited by Proceedings of the 4th International Conference on Neural Information Processing Systems. San Francisco, CA: Morgan Kaufmann Publishers, 1991. pp. 950–957
- Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR: Improving neural networks by preventing coadaptation of feature detectors. arXiv 2012. arXiv:1207.0580
- Ioffe S, Szegedy C: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv 2015. arXiv: 1502.03167
- 54. Salimans T, Kingma DP: Weight normalization: a simple reparameterization to accelerate training of deep neural networks. Edited by Proceedings of the 30th International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates, 2016. pp. 901–909
- Lei Ba J, Kiros JR, and Hinton GE: Layer normalization. arXiv 2016. arXiv:1607.06450
- 56. Li Y, Qi HZ, Dai J, Ji XY, Wei YC: Fully convolutional instance-aware semantic segmentation. Edited by Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Los Alamitos, CA: IEEE Computer Society, 2017. pp. 4438–4446
- Girshick R: Fast R-CNN. Edited by Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Los Alamitos, CA: IEEE Computer Society, 2015. pp. 1440–1448
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP: SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 2002, 16:321–357
- Krizhevsky A, Hinton GE: Using very deep autoencoders for contentbased image retrieval 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN-2011) [abstract ES2011-10], April 27—29, Bruges, Belgium; 2011
- 60. Saltz J, Gupta R, Hou L, Kurc T, Singh P, Nguyen V, Samaras D, Shroyer KR, Zhao TH, Batiste R, Van Arnam J; Cancer Genome Atlas Research Network, Shmulevich I, Rao AUK, Lazar AJ, Sharma A, Thorsson V: Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. Cell Rep 2018, 23:181–193.e7
- Bottou L: Large-scale machine learning with stochastic gradient descent. Edited by Lechevallier Y, Saporta G. In Proceedings of Compstat'2010: 19th International Conference on Computational Statistics. Berlin, Germany: Springer-Verlag, 2010. pp. 177–186

- Kingma DP, Ba J, Adam: A method for stochastic optimization. arXiv 2014. arXiv:1412.6980
- 63. Nesterov Y: A method for unconstrained convex minimization problem with the rate of convergence O (1/k²2). Dokl Akad Nauk S S S R 1983, 269:543-547
- Kim Y: Convolutional neural networks for sentence classification. arXiv 2014. arXiv:1408.5882
- Bergstra J, Bengio Y: Random search for hyper-parameter optimization. J Mach Learn Res 2012, 13:281

 –305
- Falkner S, Klein A, Hutter F: BOHB: robust and efficient hyperparameter optimization at scale. arXiv 2018. arXiv:1807.01774
- Shotton J, Winn J, Rother C, Criminisi A: TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int J Comput Vis 2009, 81:2

 –23
- 68. Krähenbühl P, Koltun V: Efficient inference in fully connected CRFS with Gaussian edge potentials. Edited by Proceedings of the 24th International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates, 2011. pp. 109–117
- 69. Iakovlev VV, Gabril M, Dubinski W, Scorilas A, Youssef YM, Faragalla H, Kovacs K, Rotondo F, Metias S, Arsanious A,

- Plotkin A, Girgis AHF, Streutker CJ, Yousef GM: Microvascular density as an independent predictor of clinical outcome in renal cell carcinoma: an automated image analysis study. Lab Invest 2012, 92:46–56
- Zhang D, Song Y, Liu D, Jia H, Liu S, Xia Y, Huang H, Cai W: Panoptic segmentation with an end-to-end cell R-CNN for pathology image analysis. Edited by Medical Image Computing and Computer Assisted Intervention — MICCAI 2018: Lecture Notes in Computer Science, vol 11071. Cham, Switzerland: Springer International Publishing, 2018. pp. 237—244
- Graham S, Vu QD, Raza SEA, Kwak JT, Rajpoot N: XY network for nuclear segmentation in multi-tissue histology images. arXiv 2018. arXiv:1812.06499
- Guillaud M, Buys TP, Carraro A, Korbelik J, Follen M, Scheurer M, Storthz KA, van Niekerk D, MacAulay CE: Evaluation of HPV infection and smoking status impacts on cell proliferation in epithelial layers of cervical neoplasia. PLoS One 2014, 9:e107088
- Ke T-W, Hwang J-J, Liu Z, Yu SX: Adaptive affinity field for semantic segmentation. arXiv 2018. arXiv:1803.10335